Delft University of Technology

Enhancing Precision Agriculture Through Human-in-the-Loop Planning and Control

Deka, Shankar A.; Phodapol, Sujet; Gimenez, Andreu Matoses; Fernandez-Ayala, Victor Nan; Wong, Rufus; Yu, Pian; Tan, Xiao; Dimarogonas, Dimos V.

DOI
10.1109/CASE59546.2024.10711319

Publication date
2024

Document Version
Final published version

Published in
Proceedings of the IEEE 20th International Conference on Automation Science and Engineering, CASE 2024

Citation (APA)
Deka, S. A., Phodapol, S., Gimenez, A. M., Fernandez-Ayala, V. N., Wong, R., Yu, P., Tan, X., & Dimarogonas, D. V. (2024). Enhancing Precision Agriculture Through Human-in-the-Loop Planning and Control. In *Proceedings of the IEEE 20th International Conference on Automation Science and Engineering, CASE 2024* (pp. 78-83). (IEEE International Conference on Automation Science and Engineering). IEEE. https://doi.org/10.1109/CASE59546.2024.10711319

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Green Open Access added to TU Delft Institutional Repository

'You share, we take care!' - Taverne project

https://www.openaccess.nl/en/you-share-we-take-care

Otherwise as indicated in the copyright section: the publisher
is the copyright holder of this work and the author uses the
Dutch legislation to make this work public.

2024 IEEE 20th International Conference on Automation
Science and Engineering (CASE)
August 28 - September 1, 2024. Bari, Italy

2024 IEEE 20th International Conference on Automation Science and Engineering (CASE) | 979-8-3503-5851-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/CASE59546.2024.10711319

Enhancing Precision Agriculture Through Human-in-the-Loop Planning and Control

Shankar A. Deka[1], Sujet Phodapol[2], Andreu Matoses Gimenez[3], Victor Nan Fernandez-Ayala[2],
Rufus Wong[2], Pian Yu[4], Xiao Tan[2] and Dimos V. Dimarogonas[2]

Abstract— **In this paper, we introduce a ROS based framework designed for the planning and control of robotic systems within the context of precision agriculture, with an emphasis on human-in-the-loop capabilities. Utilizing Linear Temporal Logic to articulate complex task specifications, our algorithm creates high-level robotic plans that are not only correct by design but also adaptable in real time by human operators. This dual-focus approach ensures that while humans have the flexibility to modify the high-level plan on-the-fly or even take over low-level control of the robots, the system inherently safeguards against any human actions that could potentially breach the predefined task specifications. We demonstrate our algorithm within the dynamic and challenging environment of a real vineyard, where the collaboration between human workers and robots is critical for tasks such as harvesting and pruning, and show the practical applicability and robustness of our software. This work marks a pioneering application of formal methods to complex, real-world agricultural environments.**

I. INTRODUCTION

Collaboration between humans and robots is becoming increasingly prevalent in today's society, wherein the precision, speed, and task repeatability of robotic systems are complemented by the adaptability, ingenuity, and context-aware decision-making capabilities of humans to achieve common objectives together. This collaboration extends well beyond industrial settings into fields such as healthcare, manufacturing, education, precision agriculture and even daily domestic activities. This paper primarily focuses on developing a framework for human robot collaboration within the context of precision agriculture, wherein teams of manipulation-endowed mobile robots perform agronomic tasks autonomously together with human co-workers that cooperate with and supervise the multi-robot team.

The agriculture sector, which traditionally has a lower technological level, is seeing a recent modernization with the rise of AI, Big Data and robotics [1]. The most widespread integration of these technologies nowadays is in the field of UAVs, which can be used to automate the process of inspection of crops [2] and to facilitate spraying of liquids, like water, fertilizer, herbicides, and pesticides [3]. One of the main benefits of increasing automation is that large teams of robots can be used to cover a larger area in a more cost-efficient manner, e.g., through drone swarms [4]. In addition, the development of specific grippers for pruning [5] and vision algorithms for the detection of complex clusters of fruits [6], together with mobile platforms capable of traversing the challenging environment of agricultural fields [7], and the recent societal interest on advancing the sector through different funding opportunities and competitions [8], have allowed researchers to start considering the more complex planning tasks related to precision agriculture.

A key requirement for enabling natural and seamless human-robot collaboration in performing complex tasks is the ability to provide abstract real-time mission specifications to the robots in a manner that is easily interpretable to the human collaborators, and yet unambiguous to the robots. Such requirements are particularly common in a precision agriculture setting, wherein robot tasks are dynamically determined by the human collaborators and communicated in real time, like possible interactions among human and robots which have to be done in a safe manner [9], immersive teleoperation [10] and social-aware robot navigation [11].

Over the past decades, Linear Temporal Logic (LTL) [12] has emerged as a conventional method for specifying the complex behavior of robotic systems. It is particularly effective since it provides rich syntax to encode complex tasks into LTL formulas that are close to natural language, facilitating their utilization, for instance, with speech-based commands. A plethora of algorithms have been developed to tackle the planning and control challenges inherent in systems governed by LTL specifications [13], alongside the development of several software toolboxes [14], [15]. Despite the significant achievements, the application of formal methods has been confined to controlled lab settings. Real-life settings like agriculture present a list of complexities surpassing those of controlled laboratory conditions. For instance, ensuring the precise localization of robots becomes more challenging due to unpredictable and dynamic surroundings. Furthermore, various constraints such as limited resources, environmental variation, and safety considerations further complicate the effective deployment of robotic systems.

In this paper, we investigate human-robot collaboration in the context of precision agriculture and the EU CANOPIES

---

*This work was supported by the the ERC CoG LEAFHOUND, the EU CANOPIES project, the Knut and Alice Wallenberg Foundation (KAW) and the Digital Futures Smart Construction project.

[1] Shankar A. Deka is with the Department of Electrical Engineering and Automation, School of Electrical Engineering, Aalto University, 02150 Espoo, Finland (Email: `shankar.deka@aalto.fi`).

[2] Sujet Phodapol, Victor Nan Fernandez-Ayala, Rufus Wong, Xiao Tan, and Dimos V. Dimarogonas are with the Division of Decision and Control Systems, School of EECS, Royal Institute of Technology (KTH), 100 44 Stockholm, Sweden (Email: `sujet, vnfa, rcywong, xiaotan, dimos@kth.se`).

[3] Andreu Matoses Gimenez is with the Department of Cognitive Robotics, Faculty of Mechanical Engineering, TU Delft, 2628 CD Delft, Netherlands (Email: `A.MatosesGimenez@tudelft.nl`).

[4] Pian Yu is with the Department of Computer Science, University of Oxford, OX1 2JD Oxford, UK (Email: `pian.yu@cs.ox.ac.uk`).

project [16], where LTL is proposed to concisely specify the desired agronomic tasks. It is built upon our previous work [15], [17], where essential adaptations are proposed to deal with the practical implementation challenges imposed by the field. Through our investigation, we seek to overcome the challenges posed by real-world scenarios and propose strategies to enhance the applicability and robustness of LTL-based approaches. To the best of our knowledge, this is the first successfull application of formal methods to complex field environments.

II. PRELIMINARIES AND PROBLEM FORMULATION

A. Linear Temporal Logic

LTL formulas over a set of atomic propositions (boolean variables) AP are recursively defined as $\varphi ::= \top \mid a \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \bigcirc \varphi \mid \varphi_1 U \varphi_2$, where $\top :=$ true, $a \in AP$, $\bigcirc$ (next) and U (until). Other useful operators are the Boolean connector disjunction $\vee$, $\Rightarrow$ (implication) and temporal operators $\Diamond$ (eventually) and $\Box$ (always). An LTL formula can be evaluated to be true or false over an infinite-length word made of subsets of AP, i.e., over $\sigma = \sigma_0\sigma_1 ... \in (2^{AP})^{\omega}$. To see the full syntax and semantics of LTL, as well as the derivation of the operators see e.g., [12].

Definition 1. *[18] A nondeterministic Büchi automaton (NBA) is a tuple $\mathcal{B} = (S, S_0, 2^{AP}, \delta, \mathcal{F})$, where S is a finite set of states, $S_0 \subseteq S$ is the set of initial states, 2^{AP} is the input alphabet, $\delta : S \times 2^{AP} \rightarrow 2^S$ is the transition function, and $\mathcal{F} \subseteq S$ is the set of accepting states.*

An infinite run s of a NBA is an infinite sequence of states $s = s_0 s_1 ...$ generated by an infinite sequence of input alphabets $\sigma = \sigma_0\sigma_1 ... \in (2^{AP})^{\omega}$, where $s_0 \in S_0$ and $s_{k+1} \in \delta(s_k, \sigma_k), \forall k \geq 0$. An infinite run s is accepted by $\mathcal{B}$ if and only if $\text{Inf}(s) \cap F \neq \emptyset$, where $\text{Inf}(s)$ is the set of states that appear in s infinitely often. There exist fast translation tools to convert LTL formulas to NBAs [19].

B. Agent Model and Task Specification

Consider a multi-agent system with N agents indexed by $\mathcal{I} = \{1, 2, 3, ..., N\}$. The dynamics of agent i is given by $\dot{x}_i = f_i(x_i, u_i)$, where x_i and u_i are the state and input of agent i, respectively. Denote by W the workspace of the multi-agent system. Then, the agent model is created by discretizing the workspace W into regions, each with allowable transitions. These transitions are constructed using the agent's dynamics f_i. To capture the transitions between regions as well as other types of region-specific actions such as "pick/drop an object", we propose to abstract the dynamics of each agent $i \in \mathcal{I}$ using a weighted Finite Transition System (wFTS) $\mathcal{T}_i$. A wFTS [20] is a tuple $\mathcal{T}_i = (Q_i, \Sigma_i, \rightarrow_i, q_0^i, AP_i, L_i, W_i)$ where $Q_i = \{q_1, ..., q_{N^t}\}$ is the finite set of states; Σ_i is the set of actions; $\rightarrow_i \subseteq Q_i \times \Sigma_i \times Q_i$ is the transition function; q_0^i is the set of initial states; AP_i is the set of atomic propositions for $\mathcal{T}_i$; $L_i : Q_i \rightarrow 2^{AP_i}$ is the labeling function; $W_i : Q_i \times \Sigma_i \times Q_i \rightarrow \mathbb{R}^+$ is the weight function as cost of transition in $\rightarrow_i$.

To allow for region-specific actions, guard functions G_i are used to identify allowable transitions in the wFTS, such that $(q_i, \sigma_i, q_i') \in \rightarrow_i$ if and only if $G_i(q_i, \sigma_i) = \top$.

The high-level agronomic tasks are represented as LTL formulas $\varphi = \varphi^{hard} \wedge \varphi^{soft}$, where φ^{hard} are the hard tasks that should be strictly satisfied, such as safety requirements for obstacle avoidance, and φ^{soft} are the soft tasks which are optional and can be violated if needed. The LTL formulas are then converted into a hard $\mathcal{B}_{\varphi^{hard}}$ and soft $\mathcal{B}_{\varphi^{soft}}$ Büchi automatons which are combined as $\mathcal{B}_{\varphi} = \mathcal{B}_{\varphi^{hard}} \times \mathcal{B}_{\varphi^{soft}}$. A parameter $\beta \in \mathbb{R}_{\geq 0}$ is used to define the penalty for violating the soft tasks [21].

C. Objectives

In our previous work [15], a ROS software package has been designed for human-in-the-loop (HIL) planning and control under LTL specifications. It is worth noting that this package was primarily designed for controlled laboratory environments. There are several challenges to implement it to complex, real-world agricultural environments. Workspace discretization from maps based on SLAM may be imperfect due to inaccuracies in GPS systems and LiDAR sensors. Communication issues can arise due to the limited network bandwidth, making it unreliable for robots to share location information. Finally, the planner divides the field into specific regions, but localization inaccuracies sometimes place the robot outside these areas, preventing planner initiation.

Our goal here is to design a ROS based framework for the planning and control of robotic systems in the domain of precision agriculture. We adopt a HIL approach, enabling humans to both adjust the high-level plan in real-time and assume direct control over the robots' low-level functions. Our aim is to 1) develop an approach for each agent i, which is capable of reacting to human inputs—even those that may pose potential risks—while ensuring safety and LTL task satisfaction and 2) engineer the ROS package to not only generate plans and control mechanisms for each agent i, but also to effectively tackle the aforementioned challenges commonly encountered in agricultural environments.

III. MULTI-ROBOT TASK PLANNING WITH HUMANS IN THE SHARED WORKSPACE

A. Methodology

The overall framework consists of high-level motion planning, low-level navigation, and online replanning. Conceptually, the high-level motion planning takes in user-specified robot tasks (LTL formula, and later translated to a NBA) and robot models (wFTS), and generates a high-level plan and action sequence based on the product automaton from the NBA and the wFTS. These high-level actions are then implemented by low-level navigation stack. In case of derivations from original plans or unforeseen scenarios, the online planning is activated that yields a new correct plan.

1) High-level motion planning: Initially, an LTL-satisfying plan needs to be generated for each robot i. Given the wFTS $\mathcal{T}_i$ for each agent $i \in \mathcal{I}$ and the Büchi automaton

$\mathcal{B}_\varphi$, the discrete plan for i is obtained by first computing the Product Büchi Automaton (PBA) $\mathcal{A}^i_{\mathcal{P}}$ defined as

$$\mathcal{A}^i_{\mathcal{P}} = \mathcal{B}_\varphi \otimes \mathcal{T}_i = \left(S^i_{\mathcal{P}}, \delta^i_{\mathcal{P}}, S^i_{\mathcal{P},0}, \mathcal{F}^i_{\mathcal{P}}, W^i_{\mathcal{P}}\right),$$

where $S^i_{\mathcal{P}} = S \times Q_i$; $(\langle s, q\rangle, \langle s', q'\rangle) \in \delta^i_{\mathcal{P}}$ iff $\exists \sigma \in \delta$, $s' \in \delta(s, \sigma)$ and $\exists \sigma_i \in \Sigma_i, (q, \sigma_i, q') \in \to_i$; $S^i_{\mathcal{P},0} = S_0 \times q^i_0$ is the set of initial states; $\mathcal{F}^i_{\mathcal{P}} = \mathcal{F} \times Q_i$ is the set of accepting states; $W^i_{\mathcal{P}} : \delta^i_{\mathcal{P}} \to \mathbb{R}^+$, $W^i_{\mathcal{P}}(\langle s, q\rangle, \langle s', q'\rangle) = W_i(q, q')$.

Afterwards, a Dijkstra-based algorithm with the PBA $\mathcal{A}^i_{\mathcal{P}}$ is used to find an optimal infinite run [22] and project it back to the wFTS $\mathcal{T}_i$ using model-checking methods [23]. The runs $r_{\mathcal{P}} = p_0 p_1 ... p_k (p_{k+1} ... p_n p_k)^\omega$ have a prefix-suffix structure, where the prefix is executed only once from the initial state $p_0 \in S^i_{\mathcal{P},0}$ to the accepting state $p_k \in \mathcal{F}^i_{\mathcal{P}}$, and the suffix is repeated infinitely from the accepting state p_k to itself.

2) Low-level navigation stack: The navigation and localization software stack provided by Università Roma Tre—which is not yet published—was used to implement the high-level actions related to motion provided by the LTL planner. The stack provides localization capabilities using the onboard Lidars with landmark-based SLAM, and navigation with a Model Predictive Control (MPC) motion planner for reaching positions in the field while avoiding collisions. The navigation stack also provides control subroutines for special cases, such as docking with another robot to exchange boxes.

A static map was generated by the navigation and localization stack that contains information about static obstacles and relevant landmarks, and was later used by the LTL planner. The algorithm also returns the location of the robot with respect to the field reference frame during runtime for the LTL planner to use.

The MPC controller takes into account additional low-level locomotion-related constraints, such as minimizing rotations to avoid marking the terrain excessively and keeping a safe distance from the plants. The controller then calculates the trajectory for a given goal position, and returns status information.

3) Online replanning: Note that the initial plan for each robot i does not account for the motion of other robots and humans in the environment. Thus, online replanning is necessary during the implementation. Each robot i can identify conflicts using the onboard Lidar sensors. Whenever a possible collision is detected, the LTL planner first conducts replanning, which finds a new accepting run for the PBA $\mathcal{A}^i_{\mathcal{P}}$ that is collision-free. Then the MPC controller inside the navigation stack performs a collision avoidance maneuver which brings the robot to the new planned trajectory.

For the HIL context, we consider that the humans can 1) specify new tasks to the robot in real time and 2) take over the control of the robot. The former is achieved by assigning appropriate arguments/options when bringing up the system and choosing among a set of pre-specified formulas. The latter is achieved through a low-level scheme based on mixed-initiative control (MIC) formulation [17], such that

$$u \triangleq u_r(x) + k(x) u_h(t), \tag{1}$$

where $u_r(x)$ is the planner input at state x, $u_h(t)$ is the human input at time t, and $k \in [0, 1]$ is a smooth function indicating how close x is to a trap state. A $trap\ state$ is defined as a PBA state from which the Büchi acceptance condition cannot be fulfilled. The set of all trap states for an agent i is denoted by $\mathcal{O}^i_t$, which has an associated distance metric to track the proximity of the current state to $\mathcal{O}^i_t$ for $\mathcal{T}_i$. To ensure that the MIC does not violate any hard task φ_{hard}, the new input must guarantee that the agent never reaches $\mathcal{O}^i_t$ no matter the human input provided. The function $k(x)$ approaches 0 (and the autonomous control takes over) when the state x is in proximity to $\mathcal{O}^i_t$, and $k(x) \to 1$ (and the human input is also applied) otherwise. To implement these inputs, the navigation stack also includes a joystick functionality that allows a human to drive the robot through direct commands.

B. Software Architecture

To use the LTL planning capabilities with the robots, the ROS LTL automaton package from [15] was used. This package requires a state machine that defines the transition system of the robot in the vineyard setting and an LTL formula that defines the tasks. The LTL formula accepts both hard and soft constraints on the tasks.

The LTL planner then creates the corresponding Büchi automatons automatically and finds a prefix and suffix plan if it exists. The planner keeps track of the plan's current state and provides the next symbolic action to take to move to the next planned state. Once the prefix plan is completed, the suffix plan will be repeated until terminated. The planner will trigger an online re-planning if the current state of the robot deviates from the predicted transition. A separate node was used to map the observations from the localization and navigation stack, as well as any other onboard sensors that detect the state of the robot, to the symbolic states defined in the transition system of the planner. The symbolic actions provided by the planner can then be interfaced and executed by the appropriate hardware and its controller.

To generate the workspace W, the robots must first explore the vineyard and create a static map using the localization software stack. Afterwards, the terrain is discretized in a grid of a size specified by the user and the allowed transitions between regions are calculated accounting for obstacles that the static map provides. These transitions are $goto\, r$ actions, which allow the robot to go from its current grid state to grid state r.

Additionally, a transition system for the robot load state is created (please see Fig. 1). In order to encode the relationship between the different transition systems, a guard G_i can be specified for the actions, i.e. the action can only be performed in a specific state of W. For example, the action *get grapes* can only be used in regions where vines with grapes are located. The total transition system that is then fed into the planner is the product transition system of both the robot load state and the workspace grid state. All these components and dependencies can be seen in Fig. 2.

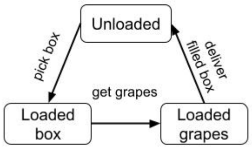

Unloaded
pick box
deliver filled box
get grapes
Loaded box
Loaded grapes

Fig. 1. Diagram of the states and actions for the robot load transition system in the vineyard.

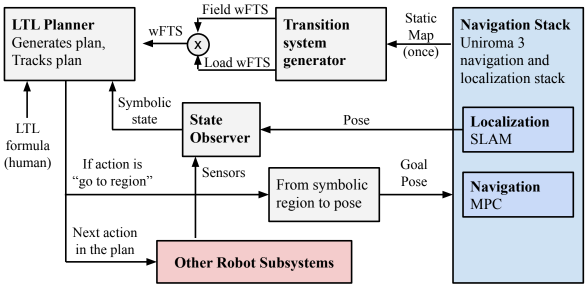

LTL Planner
Generates plan,
Tracks plan
wFTS
Field wFTS
Load wFTS
Transition system generator
Static Map (once)
Navigation Stack
Uniroma 3 navigation and localization stack
LTL formula (human)
Symbolic state
State Observer
Pose
Localization
SLAM
If action is "go to region"
Sensors
From symbolic region to pose
Goal Pose
Navigation
MPC
Next action in the plan
Other Robot Subsystems

Fig. 2. Diagram of the software stack, its dependencies, inputs, and outputs.

IV. FIELD EXPERIMENTS

This section presents a real-world application use case [24] for the LTL task planner described in the previous section. We successfully demonstrate key aspects of our planner through field experiments involving human-multirobot cooperation within the context of precision agriculture.

A. *Robotic task planning for precision agriculture*

This research was carried out as a part of the H2020 European project CANOPIES [16], with an overarching aim to design an integrated system in the field of precision agriculture for permanent crops where farm workers can efficiently collaborate with multi-robot teams to perform agronomic interventions such as harvesting or pruning in table-grape vineyards. These agronomic tasks are specified to manipulation endowed mobile robots through temporal logic syntax, which accordingly plan their action sequences in real time while accounting for dynamic entities such as humans and other robots. Besides acting as moving obstacles, the humans also collaborate with the robots in sub-tasks such as loading/unloading boxes from the robots. Furthermore, humans can specify new tasks to the robots in real time.

B. *Experimental setup*

The experiment field is shown in the bottom panels of Fig. 3. The vine stems and pergola support poles are placed in a uniform lattice, with a spacing of approximately $2.8m$. Prior to our experiments, the robots generate a 2d map of this workspace using SLAM techniques, which is then used to discretize the field as shown in the top panel of Fig. 3. The regions in the vineyard for performing agronomic actions such as *pick empty box*, *get grapes*, and *deliver filled box* are identified a priori and the corresponding cells are marked on the 2d grid map ("r45", "r43", and "r1", respectively).

We use two mobile robots for our experiments, each of which is equipped with two LIDAR sensors, GPS antennas, IMU sensors (please refer to Fig. 4(a)), which are utilized by the SLAM and navigation modules onboard each robot. The navigation module in particular interacts directly with our LTL task planner, providing it with odometry information, as well as executing the high-level commands generated by the planner as outlined in Fig. 4(b). The SLAM and navigation modules run on an intel Next Unit of Computing (NUC) and the planner runs on another dedicated NUC. Additionally, these planner NUCs on each of the robots can communicate with a remote laptop via WiFi network, allowing humans to supervise as well as specify tasks to the robots in real time.

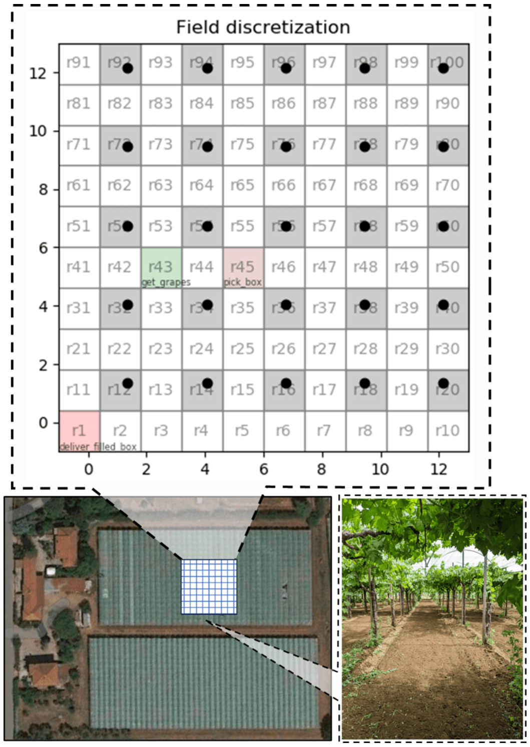

Field discretization
r91 r92 r93 r94 r95 r96 r97 r98 r99 r100
r81 r82 r83 r84 r85 r86 r87 r88 r89 r90
r71 r72 r73 r74 r75 r76 r77 r78 r79 r80
r61 r62 r63 r64 r65 r66 r67 r68 r69 r70
r51 r52 r53 r54 r55 r56 r57 r58 r59 r60
r41 r42 r43 r44 r45 r46 r47 r48 r49 r50
get_grapes
pick_box
r31 r32 r33 r34 r35 r36 r37 r38 r39 r40
r21 r22 r23 r24 r25 r26 r27 r28 r29 r30
r11 r12 r13 r14 r15 r16 r17 r18 r19 r20
r1 r2 r3 r4 r5 r6 r7 r8 r9 r10
deliver filled box
0
2
4
6
8
10
12

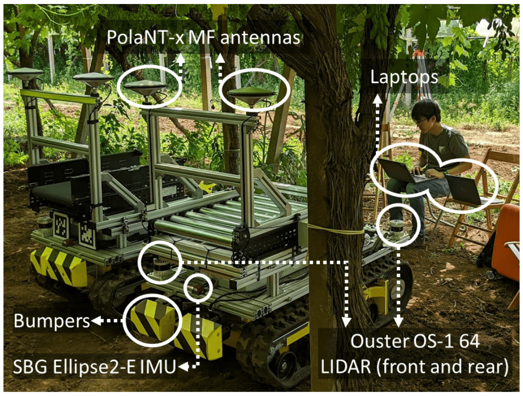
Fig. 3. Vineyard in Aprilia (Italy) where the field experiments were conducted for validation of the LTL planner. The vineyard workspace is discretized into a 10×10 grid. Each white cell is considered as a state. The grey cells contain static obstacles marked by black dots (i.e. pergola poles that support the vines as seen in the bottom right panel) and are excluded from the state transition map. The colored cells (r1, r43, r45) represent the only regions in which the specified load action can be performed.

C. *Results*

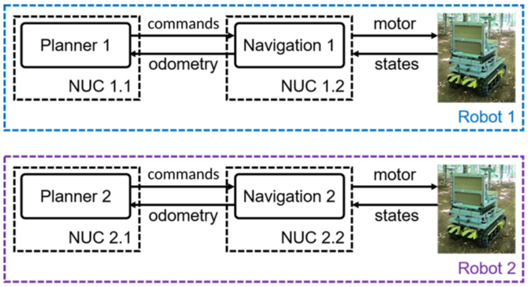
We assign tasks to each of the robots via LTL formulas as follows

$$\begin{aligned} \text{Robot 1}: &\quad \big(\Box\Diamond \text{loaded grapes}\big) \wedge \big(\Box\Diamond \text{unloaded}\big) \\ \text{Robot 2}: &\quad \big(\Box\Diamond \text{r11}\big) \wedge \big(\Box\Diamond \text{r51}\big), \end{aligned}$$

which respectively translate to "always eventually load grapes and always eventually unload grapes," and "always eventually go to region r11 and region r51." The initial state of robot 1 is (r23, "unloaded") whereas robot 2 starts at state (r61, "unloaded"). As the robots perform these tasks, a human in the shared workspace interacts with the robots in different ways: firstly by acting as a dynamic obstacle (Fig. 5(a), and secondly, by collaborating in agronomic sub-tasks of loading and unloading box (Fig. 5(b)).

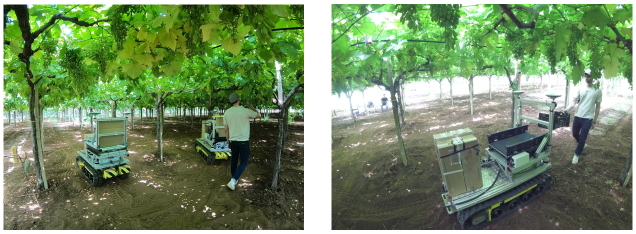


(a) Experiment hardware (other components like NUCs and routers are securely placed inside the robots.)



(b) Core modules within the robot software stack. The SLAM module is run only once at the beginning of the experiment.

Fig. 4. Equipment description.



(a) (b)

Fig. 5. Human-robot interaction scenarios in the experiment. (a) The human in the shared workspace acts as a dynamic obstacle and activates safety controllers in the robots, forcing them to perform collision avoidance maneuvers and replan. (b) The human collaborates with robot 1 in loading the box, wherein the robot reaches the loading station and awaits human action, following which it continues with rest of its assigned task.

The LTL plan generated for robot 1 is as follows: **Prefix plan:** $goto\,r24 \rightarrow goto\,r25 \rightarrow goto\,r35 \rightarrow goto\,r45 \rightarrow pick\,box \rightarrow goto\,r44 \rightarrow goto\,r43 \rightarrow get\,grapes \rightarrow goto\,r33 \rightarrow goto\,r23 \rightarrow goto\,r13 \rightarrow goto\,r3 \rightarrow goto\,r2 \rightarrow goto\,r1 \rightarrow deliver\,filled\,box \rightarrow goto\,r11$. **Suffix plan:** $goto\,r21 \rightarrow goto\,r22 \rightarrow goto\,r23 \rightarrow goto\,r24 \rightarrow goto\,r25 \rightarrow goto\,r35 \rightarrow goto\,r45 \rightarrow pick\,box \rightarrow goto\,r44 \rightarrow goto\,r43 \rightarrow get\,grapes \rightarrow goto\,r33 \rightarrow goto\,r23 \rightarrow goto\,r13 \rightarrow goto\,r3 \rightarrow goto\,r2 \rightarrow goto\,r1 \rightarrow deliver\,filled\,box \rightarrow goto\,r11$.

Similarly, the LTL plan generated for robot 2 is as follows: **Prefix plan:** $goto\,r51 \rightarrow goto\,r41 \rightarrow goto\,r31 \rightarrow goto\,r21 \rightarrow goto\,r11 \rightarrow goto\,r21 \rightarrow goto\,r31 \rightarrow goto\,r41 \rightarrow goto\,r51 \rightarrow goto\,r41$. **Suffix plan:** $goto\,r31 \rightarrow goto\,r21 \rightarrow goto\,r11 \rightarrow goto\,r21 \rightarrow goto\,r31 \rightarrow goto\,r41 \rightarrow goto\,r51 \rightarrow goto\,r41$.

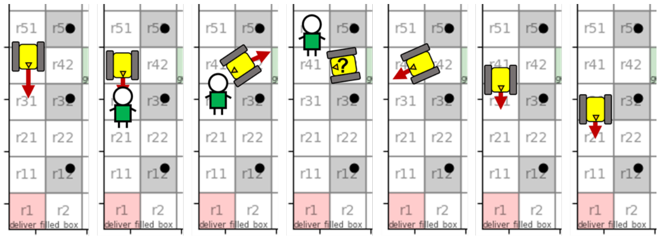


Fig. 6. The sequence of the robot motion to avoid a human in the field. Due to the interaction, the robot ends up in the region outside of the plan, $r42$. Thus, it replans and executes the new LTL plan to achieve the given task.

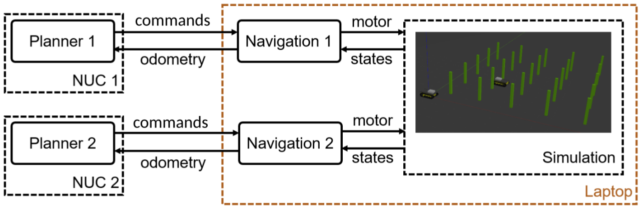


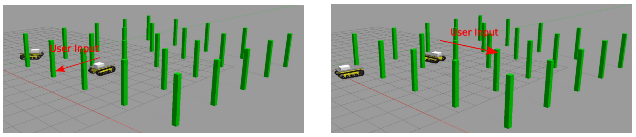
Fig. 7. Robot software stack with simulator.

As both the robots follow their respective plans, the planner also keeps track of the robot states in real time to monitor any possible deviation from the planned sequence of actions. Deviations may arise, for example, when the robot faces an obstacle and is forced to perform a safety maneuver to avoid collisions as shown in Fig. 5(a). In such a scenario, when the robot transitions to a state different from their intended state, an online re-planning is triggered and the LTL planner generates a new action sequence starting from the current state of the robot. Our experiment demonstrates this online re-planning in robot 2. As illustrated in Fig. 6, while the robot 2 is at “r41”, it is expected to transition to “r31” through the action *goto r31* as per its LTL plan. However, when the human moves towards the robot, the collision avoidance controller within the navigation module drives the robot to state “r42.” The mismatch between the expected and current state leads to a re-planning, which ultimately steers the robot 2 back to its original track once the human has safely moved away. For more details, please refer to the video in [24].

D. Gazebo Simulator

The field experiments were also reproduced in a Gazebo [25] based simulator. Gazebo is a physics-based simulator that can be used to simulate sensors and actuators and their interaction with the environment. The simulation is set up with a virtual vineyard with green pillars simulating grape vines. The robot is simulated as a simple differential drive robot with cylindrical drive wheels and two caster wheels. The LIDAR and IMU sensors are also simulated.

For the software stack, the simulation uses the same LTL formula, LTL planner, network configuration and low level navigation stack. The LTL planners runs on independent and identical NUCs as in the field experiment. The navigation



(a) (b)

Fig. 8. Two instances of human input provided through the MIC module developed in the Gazebo simulator. The human commands the low level controller to go in the direction of the red arrow while avoiding any $trap\ state$. (a) First instance. (b) Second instance.

stack and simulated hardware run on a separate computer. This mimics the network structure from the LTL planner's point of view. The LTL planner receives and sets identical data to and from the navigation stack as in the field experiment. This is illustrated in Fig. 7.

The simulator was used to test the HIL MIC module shown in (1) to ensure the system does not enter any of the $trap\ states$. Fig. 8 and the video [24] shows two instances where a human low level command is inputted through the command velocity and the MIC module provides the desired motion while ensuring that no $trap\ state$ is reached. The MIC module works in tandem with the navigation stack to prevent collisions which is also considered a $trap\ state$. When the human input stops, the LTL planner replans and the robots resume their original task.

V. CONCLUSION

In this paper, we presented a ROS based framework tailored for the advanced planning and control of HIL multi-agent systems in precision agriculture. Through LTL encoding of agronomic tasks, our system generates correct-by-design high-level robotic plans. In addition, we allow human operators to dynamically adjust these plans or assume direct control while ensuring adherence to essential task specifications. Empirical validations were conducted within the complex context of a real vineyard, where we demonstrated the practical viability, efficiency and robustness of our approach for agricultural operations and human-robot collaboration, ensuring that both predefined objectives and emergent human preferences are optimally balanced.

REFERENCES

[1] Y. Liu, X. Ma, L. Shu, G. P. Hancke, and A. M. Abu-Mahfouz, "From industry 4.0 to agriculture 4.0: Current status, enabling technologies, and research challenges," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4322–4334, 2021.

[2] P. Tripicchio, M. Satler, G. Dabisias, E. Ruffaldi, and C. A. Avizzano, "Towards smart farming and sustainable agriculture with drones," in *2015 International Conference on Intelligent Environments*, 2015, pp. 140–143.

[3] M. Shahrooz, A. Talaeizadeh, and A. Alasty, "Agricultural spraying drones: Advantages and disadvantages," in *2020 Virtual Symposium in Plant Omics Sciences (OMICAS)*, 2020, pp. 1–5.

[4] M. Liang and D. Delahaye, "Drone fleet deployment strategy for large scale agriculture and forestry surveying," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 4495–4500.

[5] E. Navas, R. Fernández, D. Sepúlveda, M. Armada, and P. Gonzalez-de Santos, "Soft gripper for robotic harvesting in precision agriculture applications," in *2021 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 2021, pp. 167–172.

[6] C. Smitt, M. Halstead, P. Zimmer, T. Läbe, E. Guclu, C. Stachniss, and C. McCool, "Pag-nerf: Towards fast and efficient end-to-end panoptic 3d representations for agricultural robotics," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 907–914, 2024.

[7] S. S. H. Hajjaj and K. S. M. Sahari, "Review of agriculture robotics: Practicality and feasibility," in *2016 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)*, 2016, pp. 194–198.

[8] S. M. Luglio, M. Sportelli, C. Frasconi, M. Fontanelli, M. Matteucci, G. Fontana, E. Piazza, and D. Facchinetti, "Field campaign and experimental design for robot performance evaluation (acre 2023)," in *2023 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor)*, 2023, pp. 398–403.

[9] P. Baxter, G. Cielniak, M. Hanheide, and P. From, "Safe human-robot interaction in agriculture," in *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 59–60. [Online]. Available: https://doi.org/10.1145/3173386.3177072

[10] Y. Chen, B. Zhang, J. Zhou, and K. Wang, "Real-time 3d unstructured environment reconstruction utilizing vr and kinect-based immersive teleoperation for agricultural field robots," *Computers and Electronics in Agriculture*, vol. 175, p. 105579, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169920311479

[11] A. Anagnostis, L. Benos, D. Tsaopoulos, A. Tagarakis, N. Tsolakis, and D. Bochtis, "Human activity recognition through recurrent neural networks for human–robot interaction in agriculture," *Applied Sciences*, vol. 11, no. 5, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/5/2188

[12] C. Baier and J.-P. Katoen, *Principles of Model Checking*. The MIT Press, Apr. 2008.

[13] C. Belta, B. Yordanov, and E. A. Gol, *Formal methods for discrete-time dynamical systems*. Springer, 2017, vol. 89.

[14] M. Guo and M. M. Zavlanos, "Probabilistic motion planning under temporal tasks and soft constraints," *IEEE Transactions on Automatic Control*, vol. 63, no. 12, pp. 4051–4066, 2018.

[15] R. Baran, X. Tan, P. Varnai, P. Yu, S. Ahlberg, M. Guo, W. S. Cortez, and D. V. Dimarogonas, "A ros package for human-in-the-loop planning and control under linear temporal logic tasks," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, 2021, pp. 2182–2187.

[16] CANOPIES project, "A Collaborative Paradigm for Human Workers and Multi-Robot Teams in Precision Agriculture Systems," in *https://canopies.inf.uniroma3.it/*, european Commission under the H2020 Framework Programme. [Online]. Available: https://canopies.inf.uniroma3.it/.

[17] M. Guo, S. Andersson, and D. V. Dimarogonas, "Human-in-the-loop mixed-initiative control under temporal tasks," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE Press, 2018, p. 6395–6400. [Online]. Available: https://doi.org/10.1109/ICRA.2018.8460793

[18] J. R. Büchi, *On a Decision Method in Restricted Second Order Arithmetic*. Springer New York, 1990, pp. 425–435. [Online]. Available: https://doi.org/10.1007/978-1-4613-8928-6_23

[19] P. Gastin and D. Oddoux, "Fast ltl to büchi automata translation," in *Computer Aided Verification*. Springer Berlin Heidelberg, 2001, pp. 53–65.

[20] M. Guo, K. H. Johansson, and D. V. Dimarogonas, "Motion and action planning under ltl specifications using navigation functions and action description language," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 240–245.

[21] M. Guo and D. V. Dimarogonas, "Multi-agent plan reconfiguration under local ltl specifications," *The International Journal of Robotics Research*, vol. 34, no. 2, pp. 218–235, 2015. [Online]. Available: https://doi.org/10.1177/0278364914546174

[22] S. L. Smith, J. Tůmová, C. Belta, and D. Rus, "Optimal path planning for surveillance with temporal-logic constraints," *The International Journal of Robotics Research*, vol. 30, no. 14, pp. 1695–1708, 2011. [Online]. Available: https://doi.org/10.1177/0278364911417911

[23] C. Belta, B. Yordanov, and E. A. Gol, *Formal Methods for Discrete-Time Dynamical Systems*. Springer International, Mar. 2017.

[24] S. A. Deka, S. Phodapol, A. Matoses Gimenez, V. Nan Fernandez-Ayala, R. Wong, P. Yu, X. Tan, and D. V. Dimarogonas. Ltl planning and control for multi-agent systems in real-world environments. Youtube. [Online]. Available: https://youtu.be/4PO25VWl6y0

[25] Gazebo, "Open Source Robotics Foundation," http://gazebosim.org/, 2014, [Online; accessed 26-Feb-2024].